



To: Mark Weiss, Associate Director, SCIS

From: SCIS Undergraduate Committee:

Rick Blazek, Debra Davis, Tim Downey, Radu Jianu, Pat McDermott, Norman Pestaina,
Nagarajan Prabakar (Chair)

Date: 2/15/16

The SCIS Undergraduate Committee has completed its review of the 2013-2015 Annual Assessment Summary for BS in CS. The committee's recommendations are included in the attached document.

UGC Recommendations on the 2013-2015 Annual Assessment Summary

In this document, the relevant sections of the Annual Assessment Summary are reproduced (in italics) and the UGC recommendations are highlighted below the sections. The following acronyms are used in this report:

AAS – Annual Assessment Summary

AC – Assessments Coordinator

SA – Subject Area

SAC – Subject Area Coordinator

SCIS – School of Computing and Information Sciences

UGC – Undergraduate Committee

I. RECOMMENDATIONS

A. Recommendations of the Subject Area Coordinators

Subject Area: Professional Development (SAC: Rick Blazek) (See AAS page 47)

ENC 3249: No changes are recommended. However, emphasis on technical writing skills should be renewed in this course.

Communicate with English Department to emphasize on technical writing skills as described in the syllabus.

Subject Area: Computer Organization (SAC: Nagarajan Prabakar) (See AAS page 47)

CDA 3103: From instructor course appraisals, students seem to be deficient in algorithmic process, basic logic and programming skills. These deficiencies need to be addressed in introductory CS courses.

Refer to the recommendation for COP 2210 (SA: Programming).

Subject Area: Foundations (SAC: Xudong He) (See AAS page 48)

COT 3541: Two instructors who taught this course noted that the students did not have adequate preparation (it was between “deficient” and “non-existent”) for the class. One commented on the continual deterioration of student quality and lack of motivation. Another commented on the students’ lack of understanding of induction and essential concepts of propositional logic that mandated sacrificing the coverage of first order logic. One possible solution to address these concerns is to offer our own Discrete Math course, which covers some materials such as propositional logic and induction, thus complements COT 3541.

Suggest the SCIS “Logic for Computer Science and Math Courses” Subcommittee to evaluate the current syllabus in the context of [ACM recommended curriculum](#). If necessary, discuss with Math faculty for changes to the Discrete Math course.

Subject Area: Programming (SAC: Norman Pestaina) (See AAS page 48)

COP 2210:

- *SCIS should provide a pre-programming course focused on problem-solving and logic skills, and that introduces the algorithmic process, abstraction and some computer programming using a minimal-syntax non-production programming language and IDE. Such a course must have clearly defined learning outcomes and evaluation methodologies.*
- *SCIS should require all students enrolled in COP 2210 to complete an evaluation no later than the first week of class, and preferably earlier, in order to recommend to the student whether to continue their COP 2210 registration, or in the pre-programming course instead.*
- *The COP 2210 common syllabus should be redesigned around carefully constructed learning outcomes that direct the focus of students and instructors towards abstraction, problem solving and the algorithmic process.*
- *SCIS should rethink the objectives and delivery mode of COP 2210 to reflect the role of this class as the introduction to the study and practice of Computer Science.*

1. *Actively engage with students during the first week and make them aware of the need for abstraction and problem solving skills; perhaps through online video and self-evaluation tools.*
2. *A group of faculty needs to discuss and revise the objectives of COP 2210 and COP 3337 in concert. Please review the details in COP 3337 below.*
3. *Evaluate the existing delivery mode.*
4. *Conduct teaching evaluation for adjuncts to improve the delivery of course.*

COP 3337:

- *The COP 3337 common syllabus should be redesigned around carefully constructed learning outcomes that direct the focus of students and instructors towards abstraction, problem solving and the algorithmic process.*
- *The operational syllabi of COP 2210 and COP 3337 must be integrated to ensure a seamless transition from COP 2210 into COP 3337 for both students and instructors. This can be facilitated by various means including*
 1. *Clear articulation of learning outcomes for both COP 2210 and COP 3337.*
 2. *Clear articulation of expected programming experiences for students in both classes, including critical feedback on students' programming style and methodology:*
 3. *Common exams for all sections of COP 2210 designed to test achievement of the learning outcomes. These need not be a complete final exam, but could be, for example, a ½ hour multiple-choice quiz. This must contribute towards the student's class grade, either as part of a final exam, or as a stand-alone activity.*
 4. *Scheduled meeting(s) of instructors of both classes at least once towards the end of each semester.*

Subject Area: Software Engineering (SAC: Masoud Sadjadi) (See AAS page 50)

CEN4010:

- *Observations:*
 - *Our students expect to learn more about the real world problems and the state of the art software engineering practices being used in industry.*
 - *They do not want to be bugged down with plenty of homework assignments and extra documentations that would be of no use to them in the future.*
 - *Our professors would like our students to perform better in their groups.*
- *Recommendations:*
 - *Adopt the state-of-the-art practices of software development from industry.*
 - *Agile and more specifically, Scrum, is the solution.*
 - *Professors of this course should adopt an Agile/Scrum book.*
 - *Class lecture times should be spent more on practicing agile software engineering development than just giving lectures.*
 - *Learning by example and practice is the best way to transfer the knowledge and experience from the professor to the students.*

UGC disagrees with suggestions to narrow the focus of this course to specific industry techniques. UGC suggests that instructors adopt a state-of-the-art approach with a balance between principles and practice. The SCIS “Software Engineering Course” Subcommittee may wish to evaluate the current syllabus in the context of [ACM recommended curriculum](#).

CEN 4021: The following recommendations are made.

- *Agile/Scrum software development management should be adopted.*
- *The students from this course should be asked to manage the projects in Introduction to Software Engineering and Senior Project courses taught in the same semester.*

UGC disagrees with suggestions to narrow the focus of this course to specific industry techniques. UGC suggests that instructors adopt a state-of-the-art approach with a balance between principles and practice. The SCIS “Software Engineering Course” Subcommittee may wish to evaluate the current syllabus in the context of [ACM recommended curriculum](#).

CEN 4072: The following recommendations are made.

- *Test-driven development is one of the popular agile software development practices in industry. Students should be exposed to this approach.*
- *Debugging should stay in the syllabus as testing without debugging would not help with improving the quality of the software solution.*
- *The lectures time should be spent more on practicing the testing/debugging methods using state-of-the-art tools.*

UGC suggests that instructors adopt a state-of-the-art approach with a balance between principles and practice. The SCIS “Software Engineering Course” Subcommittee may wish to evaluate the current syllabus in the context of [ACM recommended curriculum](#).

CIS 4911: The following observations and recommendations are made by the SAC.

- *Agile software engineering, and more specifically, Scrum should be employed for all the projects in this class.*

- *Students should be better prepared for this class. In particular,*
 - *Students should better learn UML diagrams in CEN 4010 course.*
 - *Students should learn how to be a team member in a self-organizing Agile/Scrum development team.*

UGC disagrees with suggestions to narrow the focus of this course to specific industry techniques. UGC suggests that instructors adopt a state-of-the-art approach with a balance between principles and practice. The SCIS “Software Engineering Course” Subcommittee may wish to evaluate the current syllabus in the context of [ACM recommended curriculum](#).

B. Recommendations of the Assessments Coordinator (See AAS page 52)

1. Course Related:

AC-04: From the SAC reports of various courses (CDA 3103, COP 2210, and COT 3541), it is clear that the students are quite deficient in the concepts related to algorithmic process, programming, and problem solving. One way to address this issue is to provide a pre-programming course focused on problem solving and logic skills. Students in COP 2210 should be evaluated in the first week of classes in order to recommend them to enroll in this pre-programming course before taking COP 2210. [Important Note: We have already created COP 1000 but need to ascertain that it is more widely advertised to the student community through our advisers.]

Refer to the recommendation for COP 2210 (SA: Programming).

AC-06: The Subject Area Coordinator for Software Engineering recommends the adoption of the state-of-the-art practices of software development from industry. In particular, Agile and Scrum should be used in CEN 4010, CEN 4021, and CIS 4911 (student projects must use these technologies).

UGC disagrees with suggestions to narrow the focus of these courses to specific industry techniques. UGC suggests that instructors adopt a state-of-the-art approach with a balance between principles and practice. The SCIS “Software Engineering Course” Subcommittee may wish to evaluate the current syllabi in the context of [ACM recommended curriculum](#).

2. Procedure Related:

AC-10: The following is not a major issue, but is noted here for future reminder purposes more than anything else. The style of Course Embedded Assessments of different instructors for the same course is sometimes quite different. SCIS should make Subject Area Coordinators responsible to maintain the consistency of this assessment. Before designing and conducting the assessment for a particular class, instructors should consult the SAC.

Suggest Subject Area Coordinators to coordinate with faculty who teach in their subject areas and to ensure that faculty select questions from a pool of questions created for this assessment.